

This Page Is Inserted by IFW Operations
and is not a part of the Official Record

BEST AVAILABLE IMAGES

Defective images within this document are accurate representations of the original documents submitted by the applicant.

Defects in the images may include (but are not limited to):

- BLACK BORDERS
- TEXT CUT OFF AT TOP, BOTTOM OR SIDES
- FADED TEXT
- ILLEGIBLE TEXT
- SKEWED/SLANTED IMAGES
- COLORED PHOTOS
- BLACK OR VERY BLACK AND WHITE DARK PHOTOS
- GRAY SCALE DOCUMENTS

IMAGES ARE BEST AVAILABLE COPY.

**As rescanning documents *will not* correct images,
please do not report the images to the
Image Problem Mailbox.**



PATENT ABSTRACTS OF JAPAN

(11) Publication number: **05289870 A**(43) Date of publication of application: **05 . 11 . 93**

(51) Int. Cl

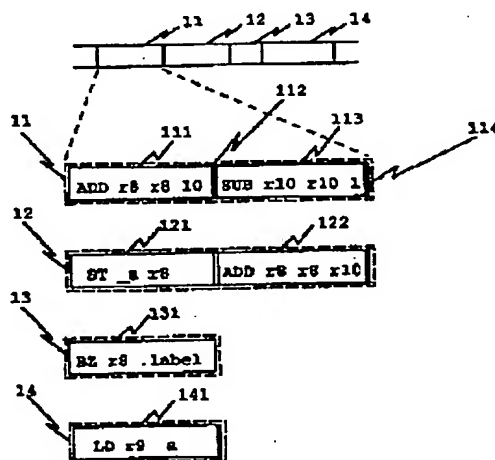
G06F 9/38**G06F 9/38**(21) Application number: **04115216**(71) Applicant: **NEC CORP**(22) Date of filing: **09 . 04 . 92**(72) Inventor: **NAKAMURA SAORI**(54) **PARALLEL INSTRUCTION EXECUTION
CONTROL SYSTEM**

COPYRIGHT: (C)1993,JPO&Japio

(57) Abstract:

PURPOSE: To easily execute the parallel execution control by providing a flag for showing the end of an instruction group in a field of an assembly instruction, and fetching and executing up to the instruction to which its flag is set at each cycle.

CONSTITUTION: When four assembly instructions, that is, 111, 113, 121 and 122 are read in, since a flag 114 is set in advance to the instruction 113, in this cycle, 111 and 113, that is, a parallel assembly instruction 11 is executed, and 121 and 122 are not executed until the next cycle. In the next cycle, since a flag is set in advance to 122, 121 and 122, that is, a parallel assembly instruction 12 is executed. In the same way, in the subsequent cycle, parallel assembly instructions 13, 14 are executed. In this case, no flag exists in an instruction 131 because it is assumed that the parallel assembly instruction is always delimited in a branch instruction.



(19)日本国特許庁 (J P)

(12) 公開特許公報 (A)

(11)特許出願公開番号

特開平5-289870

(43)公開日 平成5年(1993)11月5日

(51)Int.Cl.⁴

G 0 6 F 9/38

識別記号

3 7 0 X

庁内整理番号

9290-5B

3 1 0 X

9290-5B

F I

技術表示箇所

審査請求 未請求 請求項の数1(全 3 頁)

(21)出願番号

特願平4-115216

(22)出願日

平成4年(1992)4月9日

(71)出願人 000004237

日本電気株式会社

東京都港区芝五丁目7番1号

(72)発明者 中村 さおり

東京都港区芝五丁目7番1号 日本電気株

式会社内

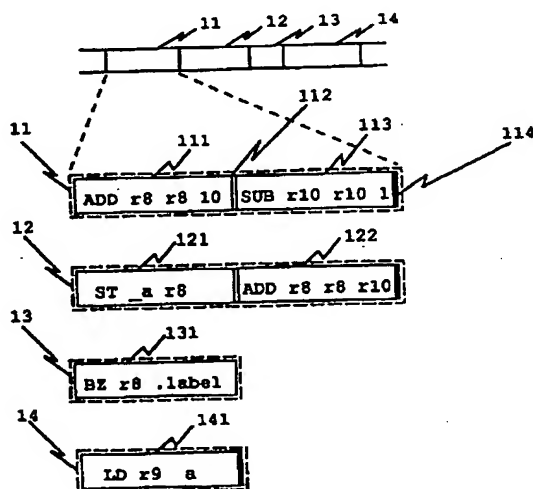
(74)代理人 弁理士 山下 穰平

(54)【発明の名称】 並列命令実行制御方式

(57)【要約】

【目的】 複雑なハードウェアを用いることなく、プロセッサの並列実行制御を容易に行えるようにする。

【構成】 複数の演算部を有するプロセッサの各々の演算部を目的プログラムであるアセンブリ命令列に含まれる各アセンブリ命令によって並列に制御する並列命令実行制御方式において、前記アセンブリ命令のフィールド内に命令群の終りであるか否かを示すフラグを設け、前記プロセッサの実行時には、サイクル毎にそのフラグがセットされている命令までを取り出して前記各演算部で並列に実行させる。



【特許請求の範囲】

【請求項1】 複数の演算部を有するプロセッサの各々の演算部を目的プログラムであるアセンブリ命令列に含まれる各アセンブリ命令によって並列に制御する並列命令実行制御方式において、前記アセンブリ命令のフィールド内に命令群の終りであるか否かを示すフラグを設け、前記プロセッサの実行時には、サイクル毎にそのフラグがセットされている命令までを取り出して前記各演算部で並列に実行させることを特徴とする並列命令実行制御方式。

【発明の詳細な説明】

【0001】

【産業上の利用分野】 本発明は、複数の演算部を有するプロセッサの各演算部を並列に制御するための並列命令実行制御方式に関するものである。

【0002】

【従来の技術】 前記のようなプロセッサにおいては、コンパイラが目的プログラムを生成する時点で命令及びデータの依存関係が解析され、その結果に基づいてアセンブリ命令の配置が行なわれる。生成される目的プログラムたるアセンブリ命令列は並列アセンブリ命令の並びで、並列アセンブリ命令の各々は演算部の数に等しい数のアセンブリ命令を含むが、一個又はそれ以上の演算部で実行すべき命令が存在しないサイクルにおいては、並列アセンブリ命令中の対応する箇所には何も実行しないことを示す空命令が置かれる。実行時にはサイクル毎に、目的プログラムたるアセンブリ命令列から1個の並列アセンブリ命令即ち演算部の数に等しい数のアセンブリ命令群を取り出してデコードし、その結果に基づいて各々の演算部が制御される。

【0003】

【発明が解決しようとする課題】 しかしながら、従来にあっては実際には各サイクルにおいて全ての演算部に対してアセンブリ命令を実行させることは不可能であることが多く、特に並列性の低いプログラムの場合、目的プログラムは多くの空命令を含むことになる。こうした問題を解決する手段として、例えば Clowell R. P., Nix R. P., O'donnell J. J., Papworth D. B. and Rodman P. K.: "A VLIW architecture for a trace scheduling compiler", IEEE Trans. Computers, 37, 8, pp. 967-979 (1988-08) に、展開後の命令の位置情報を付加して目的プログラムの圧縮を行ない、実行時に専用のハードウェアを用いてキャッシュメモリ上で圧縮された目的プログラムを展開する方式が提唱されている。ところが、この方式では展開用ハードウェアの構成が非常に複雑になるという難点がある。

【0004】 また、前記の問題がコンパイラが静的に目

的プログラムの配置を行なうことに起因するものであることから、それとは全く異なる側面からの並列実行制御の手段として、実行時に動的に命令の依存関係の解析を行ないそれに基づいて複数の演算部を持つプロセッサの実行制御を行なおうとする superscalar という方式が、McGeady S.: "The 1960 CA superscalar implementation of the 80960 architecture", Compcon Spring 90 digest of papers, pp. 232-240, IEEE (1990) 等により提案されている。しかし、この superscalar 方式では命令の依存関係及び資源の利用可能性の解析のための複雑で膨大なハードウェアが必須であり、また命令の依存関係の解析が狭い範囲に限定されるという難点があった。

【0005】 本発明の目的は、前記の問題を解消して、不要な空命令を含まないサイズの小さな目的プログラムによって、複雑なハードウェアを用いることなく前記のようなプロセッサの並列実行制御を容易に行なえるようにした並列命令実行制御方式を提供することにある。

【0006】

【課題を解決するための手段】 本発明における並列命令実行制御方式は、アセンブリ命令のフィールド内に命令群の終りを示すフラグを設けて、各サイクル毎にそのフラグがセットされている命令までを取り出して実行することにより、従来のような不要な空命令を含むことなく容易に並列実行制御を行なうことを特徴とするものである。

【0007】

【実施例】 次に、本発明の実施例について図面を参照して詳細に説明する。図1は本発明の並列命令実行制御方式に基づく目的プログラムの一実施例を示した図である。以下、図1の実施例の手順について具体的に説明する。なお、ここで対象としているプロセッサは、1サイクルに整数演算命令（またはロード/ストア命令）を2個、浮動小数点演算命令を1個、分岐命令を1個同時に実行することが可能であるとし、また簡略化のため遅延は生じないものとする。

【0008】 図1において、11, 12, 13, 14はそれぞれ1サイクルで同時に実行されるべき並列アセンブリ命令を示す。この並列アセンブリ命令は可変個のアセンブリ命令を含んでいる。例えば、11の並列アセンブリ命令は各々の整数演算器で実行されるべきアセンブリ命令111と113を含む。ここで、111はレジスタr8の内容に整数10を加算して結果をr8に置くことを表し、113はレジスタr10の内容から整数1を減算して結果をr10に置くことを表す。また、112, 114は各々命令111, 113が並列アセンブリ命令の終りか否かを示すフラグである。112はセットされていないフラグで、114はセットされているフラ

グである。同様に、12の並列アセンブリ命令は、r8の内容を変数aへストアする命令121とr8とr10の内容を加算して結果をr8へ置く命令122を含んでいる。また、13の並列アセンブリ命令はr8の内容が0と等しいならば、ラベルlabelで示されるアドレスへ分岐する命令131を含み、14の並列アセンブリ命令はaの内容をレジスタr9へロードする命令141を含むことを示している。

【0009】図1では、まず4個のアセンブリ命令、すなわち111、113、121、122が読み込まれるが、命令113はフラグ114がセットされているので、このサイクルでは111及び113すなわち並列アセンブリ命令11が実行され、121、122は次のサイクルまで実行しない。次のサイクルでは、122にフラグがセットされているので、121及び122すなわち並列アセンブリ命令12が実行される。同様にして、それ以降のサイクルで並列アセンブリ命令13、14が実行される。ここで命令131にはフラグが存在しないが、これは分岐命令では必ず並列アセンブリ命令が区切れるものと仮定しているためである。

【0010】対比のため、図2に従来方式による目的プログラムの例を示す。図2の21、22、23、24は並列アセンブリ命令を示すが、この場合1個の並列アセンブリ命令は常に4個のアセンブリ命令から構成され *

る。その4個の中にはNOPで表される空命令が1個またはそれ以上含まれることもある。例えば、並列アセンブリ命令21は211、212、213、214のアセンブリ命令を含むが、213、214は空命令である。従来方式では、サイクル毎に21、22、23、24の順に読み込まれて実行されるが、この場合半分以上が空命令であり、本発明にくらべて2倍程のメモリ及びバス等の資源を浪費することになる。

【0011】

10 【発明の効果】以上に示した如く本発明は、不要な空命令を含まない目的プログラムを用いて、前記のようなプロセッサに対する並列実行制御を容易に行なえるという効果がある。

【図面の簡単な説明】

【図1】本発明の並列命令実行制御方式に基づく目的プログラムの一実施例を示した図である。

【図2】従来方式による目的プログラムを示した図である。

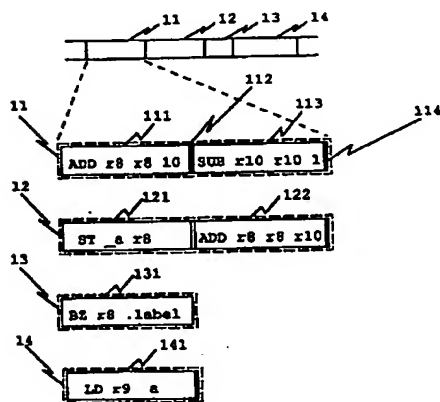
【符号の説明】

20 11~14 並列アセンブリ命令

111、113、121、122、131、141 各演算器で実行されるべきアセンブリ命令

112、114 フラグ

【図1】



【図2】

